Assessing Search Strategies for Flexible Docking

MICHAL VIETH,* JONATHAN D. HIRST, BRIAN N. DOMINY, HEIDI DAIGLER,† CHARLES L. BROOKS III

Department of Molecular Biology, TPC-6, The Scripps Research Institute, 10550 North Torrey Pines Road, La Jolla California 92037

Received 17 February 1998; accepted 16 June 1998

ABSTRACT: We assess the efficiency of molecular dynamics (MD), Monte Carlo (MC), and genetic algorithms (GA) for docking five representative ligand–receptor complexes. All three algorithms employ a modified CHARMM-based energy function. The algorithms are also compared with an established docking algorithm, AutoDock. The receptors are kept rigid while flexibility of ligands is permitted. To test the efficiency of the algorithms, two search spaces are used: an 11-Å-radius sphere and a 2.5-Å-radius sphere, both centered on the active site. We find MD is most efficient in the case of the large search space, and GA outperforms the other methods in the small search space. We also find that MD provides structures that are, on average, lower in energy and closer to the crystallographic conformation. The GA obtains good solutions over the course of the fewest energy evaluations. However, due to the nature of the nonbonded interaction calculations, the GA requires the longest time for a single energy evaluation, which results in a decreased efficiency. The GA and MC search algorithms are implemented in the CHARMM macromolecular package. © 1998 John Wiley & Sons, Inc. J Comput Chem 19: 1623–1631, 1998

Keywords: docking; genetic algorithms (GA); simulated annealing (SA); Monte Carlo (MC); molecular dynamics (MD), scoring functions

**Present address*: Computational Chemistry and Molecular Structure Research, Lilly Research Laboratories, Lilly Corporate Center, Indianapolis, IN, USA

† *Present address*: Department of Chemistry, University of Washington, Seattle, WA, USA

Correspondence to: C. L. Brooks III; e-mail: brooks@scripps.edu

Contract grant sponsor: NIH; contract/grant numbers: GM37554, RR12255

CCC 0192-8651 / 98 / 141623-09

Introduction

Lead generation is an important early step in drug discovery.[1] If the structure of the receptor is available, the major hurdles are to predict the orientation of a drug candidate in the active site and to estimate its binding affinity.[2,3] The former is addressed by molecular docking algorithms.[4] Since the introduction of DOCK,[5] the first docking program, a variety of search algorithms have been proposed to solve the docking problem,[4] including simulated annealing Monte Carlo (MC) methods,[6,7] molecular dynamics (MD),[8] and evolutionary algorithms[9] (EA). In recent years, methods involving genetic algorithms[10] (GA) from the family of EA have arguably become the dominant approach to molecular docking. The goal of this study is to compare the efficiencies of MD, MC, and GA as docking algorithms for search spaces of different sizes, using an energy function based on CHARMM,[11] which has been chosen based on its selectivity and efficiency in docking studies, as described in the preceding study[12] and elsewhere.[13] In addition, we provide an integrated toolbox for molecular docking in CHARMM that contains the energy function and different search strategies.

Westhead et al.[14] concluded that the GA is the most efficient search strategy, based on its ability to locate low energy solutions for a small search space around the active site. In contrast to that work, we base our comparison of search strategies on the efficiency of docking (i.e., the average number of correctly docked structures per unit of time). We have made every reasonable effort to optimize all search strategies for maximum efficiency on our test case of five diverse ligand–receptor complexes. It is possible that optimization using a different energy function or different implementations could lead to a different final result. So, for greater generality, we assess the search algorithms using two different sized search spaces.

We also compare our search algorithms with AutoDock,[6] one of the first docking algorithms utilizing MC-simulated annealing. It uses a very fast grid-based energy function and for the best performance frequently restarts from the lowest energy structures. AutoDock is currently a popular choice for docking applications (e.g., ref. 15, 16) and hence provides a reasonable benchmark for comparison.

In our test cases the protein receptors remain rigid, while ligand flexibility is permitted. Rigidity of receptors, in the spirit of the lock and flexible key mechanism of binding, allows for a more direct and faster comparison of the algorithms. In reality, many proteins exhibit an induced fit[17] mode of binding, but *a priori* prediction of protein rearrangement during ligand binding is, to our knowledge, not currently possible. Keeping the protein rigid can be treated as a first approximation to the general flexible docking problem and will enable us to perform a fair comparison of search strategies.

Method

Our test set comprises five ligand–receptor complexes from the Brookhaven Protein Data Bank[18] (PDB), as described in the preceding paper. It includes benzamidine/trypsin,[19] phosphocholine/FAB McPC-603,[20] sialic acid/hemagglutinin,[21] glycerol 3-phosphate/triose phosphate isomerase,[22] and biotin/streptavidin.[23] We compare the algorithms based on the geometry of the resulting docked complexes. We are not so concerned with the energetics of the solutions, because, in the preceding paper we developed an energy function such that docked structures (geometrically correct solutions) generally have lower energies than misdocked ones (geometrically incorrect solutions). However, for each structure classified as docked or partially docked (within 3 Å root-mean-square deviation [RMSD] of the crystallographic structure), we verify that its energy is lower than any of the misdocked structures (more than 4 Å RMSD from the PDB crystal structure).

The search algorithms are assessed using the efficiency metric introduced in the preceding study[12]; that is:

$$SE = \begin{cases} \frac{1}{T}\sum_{i=1}^{N}\left(f_{i,<2} + 0.5\left(f_{i,<3} - f_{i,<2}\right)\right); \\ \quad \prod_{i}^{N} f_{i,<3} \neq 0 \\ 0; \quad \prod_{i}^{N} f_{i,<3} = 0 \end{cases} \tag{1}$$

where SE is the efficiency of a given search algorithm, $f_{i,<a}$, indicates the fraction of structures with a RMSD less than a Å from the crystal

structure and an energy lower than the lowest energy misdocked structure. N is the number of complexes ($N = 5$) and T is the mean time required to dock the ligands to their receptors. The efficiency is zero if, for any complex, $f_{i,<3}$ is zero.

Based on consideration of docking selectivity and efficiency as presented in the preceding article,[12] the energy function employed here uses a $3r$ distance-dependent dielectric. A nonbonded interaction truncation at 8 Å is applied. We note that the nonbonded interaction truncation is different from the nonbonded list truncation. The latter is a cutoff that controls the creation of the atom neighbor list, which is processed using the interaction truncation of 8 Å to compute the energy and forces of the system. This neighbor list is created to diminish the cost of computing the energy and the cutoff used for list generation depends on the size of the moves. The GA leads to larger moves than the other search strategies, and thus requires a large nonbonded list truncation. A switching function was applied for electrostatics and van der Waals (vdW) interactions, with a switching distance between 6 Å and 8 Å. The number of steps/generations for each algorithm was chosen such that MC and MD have the same number of energy evaluations, and the GA showed timing similar to that of the MC. Thus, the GA will perform the lowest number of energy evaluations, because it requires the longest computational time to perform an energy evaluation in our current implementation. In what follows we describe our implementation of the GA in more detail, followed by the procedures used for all three algorithms: MC, MD, and GA. The details of the MC and MD approaches are evident in a later section.

GENETIC ALGORITHM

Since their inception in late 1970s,[24] GAs[25] and other EAs[9] have become increasingly popular as optimization tools in computer-assisted drug design. Some of the earliest applications of GAs to molecular docking are contained in the works of Dixon[10] and Judson.[26] A GA evolves the population of possible solutions through genetic operators (mutations and crossovers) to a final population, optimizing a predefined fitness function.[27]

For the purpose of flexible docking, the translational, rotational, and internal degrees of freedom are encoded into genes. Every individual (a chromosome) consists of a collection of genes and is assigned a fitness. Each chromosome is represented by a string of real numbers, each encoding a different degree of freedom. An example of a chromosomal representation of a ligand would be coding the degrees of freedom of phosphocholine (three center-of-mass coordinates, three Euler angles, and four dihedral angles) as ten genes in a chromosome. The number of genes per chromosome depends on the number of rotatable bonds for a given ligand and varies from 6 for benzamidine (three center-of-mass coordinates and three Euler angles) to 16 for sialic acid (three translational and three rotational degrees of freedom and ten internal rotatable bonds). A new population is generated from the old one by the use of three genetic operators. These genetic operators are depicted schematically in Figure 1. The mutation operator changes the value of the gene by a random value depending on the type of gene. Crossover exchanges a set of genes from one parent chromosome to another. Migration moves individual chromosomes from one subpopulation or niche to another. The niche technique has been employed to avoid premature convergence. It involves evolution of semiindependent subpopulations (five in our case) and sporadic migration of individuals between subpopulations. A new subpopulation is created from an old one by breeding parents over crossovers, mutations, and migrations and replacing all but the two most fit parents by children in every subpopulation. Thus, we use the so-called generational update with an elitism of two.[27] Parents are selected for breeding based on their relative fitness according to roulette wheel selection. A selection pressure of 1.2 is applied based on a linearly normalized ranking, following Jones et al.[28] A schematic description of GA evolution proceeds as follows: (1) random generation of initial subpopulations—each of N members; (2) ranking of each member according to fitness; (3) selection of $N - 2$ parents for breeding according to roulette wheel selection; (4) breeding of parents by the use of genetic operators (crossover or mutation or migration is chosen depending on their relative weight); (5) replacement of $N - 2$ (elitism of 2) least-fit parents in each subpopulation by children; (6) score all members in new subpopulations according to their fitness; (7) repeat procedure from (3) until the convergence or the maximum number of steps is reached.

Our real number representation of chromosomes differs only slightly from the binary representation frequently used in GA applications.[28] As

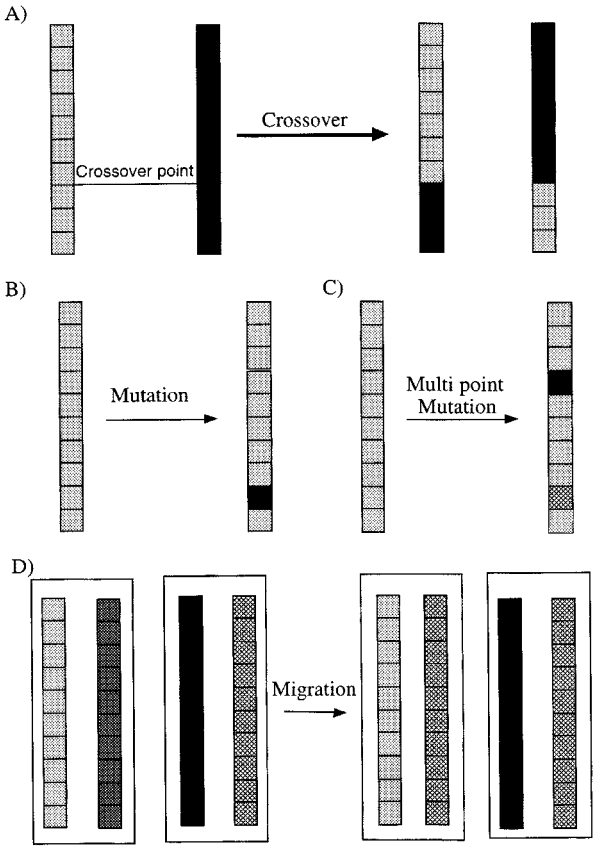

FIGURE 1. Different genetic operators used to create children chromosomes from parent chromosomes. (A) One-point crossover. Crossover point is chosen randomly, and the information between two parents is exchanged. Two children are created, each having genes from both parents. (B) Single-point mutation. A gene to be mutated is selected at random and its value is modified. (C) Multipoint mutation introduces the possibility of changing several genes at once. (D) Migration operation allows replacement of the least-fit individuals from each subpopulation by the most-fit individuals from other subpopulations.

far as crossover operators are concerned, a binary representation with crossover points between genes (no crossover points inside the genes) is equivalent to a real number representation. As pointed out by Judson,[29] allowing crossovers inside the words would introduce large random mutations, slowing the search process. With regard to the mutation operator, a binary representation is similar to a real number representation with variable size mutations. Our MC search strategy is implemented as a special case of GA with a population of independent chromosome replicas and mutations only and Boltzmann-weighted selection between parents and children.

SEARCH PROTOCOLS

All search algorithms (GA, MD, MC) share a similar protocol, which we outline in what follows. The algorithm-specific features are presented in Tables Ia, Ib, and Ic for the three stages of the protocol. To improve the performance of the individual search strategies there are some variations of details, which arise from differences in step sizes and differences in rigidity of bonds and angles (only MD uses a flexible bond/angle model). The docking protocol we use can be subdivided into three stages. The purpose of the first stage is to explore the energy surface and to find the broad energy basin where binding will occur. A soft-core nonbonded potential is employed with a nonbonded interaction reduction at $0.885\sigma_{ij}$ (σ_{ij} is the sum of the van der Waals [vdW] radii of two interacting atoms). The annealing for MC and MD starts at relatively high temperatures (900 K and 700 K, respectively) and the GA has a high mutation rate. This step accounts for about 60% of the search time. The main purpose of the second stage is to find local minima in the basins provided by the first step. We employ hardening of the soft-core nonbonded interactions at 0.6σ (0.75σ for the GA). Annealing for MC and MD starts from a lower temperature (600 K and 500 K, respectively), and the GA has a higher crossover rate. However, in some cases, new basins can also be found in this stage of the protocol. This step accounts for about 30% of the search time. In the final stages, the soft core disappears and the structures from MC and MD are quenched to a low temperature. The role of the last stage is to locally minimize structures and this accounts for about 10% of the total search time.

An initial spherical distribution of ligands around the active site is generated randomly for each GA run. Depending on the size of search space, ligands were distributed between 0.3 Å and 11 Å (or 2.5 Å as an outer boundary for the small search space) from the center of the active site. The center of the active site is defined as the location of the atom closest to the geometric center of the ligand in the PDB crystal complex. Initial ligand conformations were generated with random values of dihedral angles and Euler angles (i.e., orientations). Initial conditions for MC and MD for the large search space were, for the purpose of comparison with AutoDock, generated by AutoDock.

For the GA we first select the operation type (crossover or mutation) with a given probability. If the crossover operation is selected, all crossover

TABLE Ia.
Parameters for First Stage of Search Strategies.

Parameter	GA small	GA large	MC small	MC large	MD small	MD large
Number of steps	130	300	17,000	70,000	17,000	70,000
Simulation time (ps)	N / A	N / A	N / A	N / A	36	105
Population / number of niches	90 / 5	150 / 5	1	1	1	1
Soft core truncation	0.885σ	0.885σ	0.885σ	0.885σ	0.885σ	0.885σ
Initial temperature (K)	N / A	N / A	900	900	700	700
Final temperature (K)	N / A	N / A	400	400	300	300
Temperature change frequency	N / A	N / A	10	10	150	150
Nonbonded list update	40	40	50	25	200	200
Nonbonded list truncation (Å)	11	13.5	10.5	9	9	9
Mutation rate	0.7	0.7	N / A	N / A	N / A	N / A
Evolutionary pressure	1.2	1.2	N / A	N / A	N / A	N / A
Migration frequency	100	100	N / A	N / A	N / A	N / A
Crossover probability	0.3	0.3	0	0	0	0
Internal move probability[a]	0.6	0.6	0.6	0.6	N / A	N / A
Conditional probabilities of mutations[b]						
Dihedral angles	0.3	0.3	0.3	0.3	N / A	N / A
Rigid body degrees of freedom	0.7	0.7	0.7	0.7	N / A	N / A
One rigid body variable	0.4	0.4	0.4	0.4	N / A	N / A
Three translations or rotations	0.3	0.3	0.3	0.3	N / A	N / A
All six translations and rotations	0.3	0.3	0.3	0.3	N / A	N / A
Maximum step sizes						
Dihedral angles (°)	15	15	15	15	N / A	N / A
Translations (Å)	0.5	0.5	0.3	0.6	N / A	N / A
Rotations (°)	28	28	17	34	N / A	N / A
Big steps BS frequency[c]	20	20	50	50	N / A	N / A
Maximum step sizes for BS						
Dihedral angles (°)	20	20	30	30	N / A	N / A
Translations (Å)	0.9	0.9	0.4	0.9	N / A	N / A
Rotations (°)	52	52	22.5	52	N / A	N / A

[a] The probabilities of different modifications apply to mutations only. All crossover points have the same probabilities.
[b] The sum of the conditional probabilities at a given stage of selection is 1. If a rigid body modification is selected, we can either modify one translational or rotational degree of freedom with a probability of 0.4, three rotational or three translational degrees of freedom with probability of 0.3, or all six translational and rotational degrees of freedom with probability of 0.3.
[c] Big steps are performed to increase the exploration of the space.

points are equally probable. If mutation is chosen, we select whether an internal or rigid body degree of freedom is to be modified, depending on their relative probability. An internal move involves modification of torsion angles only. If a rigid body degree of freedom is chosen we select whether to modify all three translational or rotational degrees of freedom, all rigid body degrees of freedom, or one rigid body degree of freedom. The process of mutation selection is the same for the MC scheme.

To fine-tune solutions obtained by the GA and MC approaches, at the end of the protocol each structure is subjected to 7 ps of MD cooling from 350 K to 300 K, with a cooling frequency of 100 steps, a temperature decrement of 10 K, a nonbonded list truncation of 9 Å, and the standard nonbonded potential. Subsequently, the structures are minimized by 200 steps of conjugant-gradient minimization with an energy tolerance of 0.001 kcal/mol.

TABLE Ib.
Parameters for Second Stage, if Different from First Stage.

Parameter	GA small	GA large	MC small	MC large	MD small	MD large
Number of steps	30	100	6000	30,000	6000	70,000
Simulation time (ps)	N / A	N / A	N / A	N / A	14	45
Soft core truncation	0.75σ	0.75σ	0.6σ	0.6σ	0.6σ	0.6σ
Initial temperature (K)	N / A	N / A	600	600	500	500
Final temperature (K)	N / A	N / A	300	300	300	300
Temperature change frequency	N / A	N / A	100	100	150	150
Nonbonded list update	25	50	50	50	200	200
Nonbonded list truncation (Å)	11	13.5	9	9.5	9	9
Mutation rate	0.6	0.3	N / A	N / A	N / A	N / A
Migration frequency	25	50	N / A	N / A	N / A	N / A
Evolutionary pressure	1.1	1.1	N / A	N / A	N / A	N / A
Crossover probability	0.4	0.4	N / A	N / A	N / A	N / A
Mutation probability	0.6	0.6	1	1	1	1
Maximum step sizes						
Translations (Å)	0.25	0.4	0.2	0.4	N / A	N / A
Rotations (°)	11	23	11.5	23	N / A	N / A
Maximum step sizes for BS						
Dihedral angles (°)	20	20	20	20	N / A	N / A
Translations (Å)	0.9	0.9	0.3	0.6	N / A	N / A
Rotations (°)	52	52	17	34	N / A	N / A

TABLE Ic.
Parameters for Final Stage, if Different from First Stage.

Parameter	GA small	GA large	MC small	MC large	MD small	MD large
Number of steps	45	100	2000	8000	2000	8000
Simulation time (ps)	N / A	N / A	N / A	N / A	3	12
Soft core truncation	0.5σ	0.5σ	NO	NO	NO	NO
Initial temperature (K)	N / A	N / A	400	400	400	400
Final temperature (K)	N / A	N / A	50	50	50	50
Nonbonded list truncation (Å)	10	12.5	9	9.5	9	9
Mutation rate	0.3	0.3	N / A	N / A	N / A	N / A
Crossover probability	0.7	0.7	N / A	N / A	N / A	N / A
Mutation probability	0.3	0.3	1	1	1	1
Internal move probability	0.8	0.8	0.8	0.8	N / A	N / A
Maximum step sizes						
Dihedral angles (°)	15	15	10	10	N / A	N / A
Translations (Å)	0.25	0.25	0.1	0.2	N / A	N / A
Rotations (°)	11.5	11.5	7	11.5	N / A	N / A

For AutoDock[6] we used 1,500,000 (300,000 for the small search space) energy evaluations in 50 MC cycles. All step sizes were standard. The cubic energy grid used in all AutoDock calculations was spaced 0.375 Å between each of the 61^3 grid points, creating a square box of length 22.8 Å centered on the active site. Initial center of mass coordinates (tran0), rigid body rotation (quat0), and torsion angles (dihe0) were selected randomly. The translation step size was 3 Å, with a decrement of 0.2 Å

for each cycle. Quaternians for rigid body rotation, as well as dihedral angles, start with a 24° step size and 5° decrement per cycle. The starting temperature for each of the 100 annealing runs was 600 RT and a linear cooling schedule was used over 50 temperature reduction cycles. The minimum energy state was chosen at the end of the run. The maximum number of accepted and rejected MC steps was 25,000. An external grid energy of 1000 kcal and a maximum initial energy of 0 kcal were used to create reasonable starting positions. Internal electrostatic energies were calculated.

Results

LARGE SEARCH SPACE

The performances of the different search algorithms in the large search space are compared in Table II. The efficiencies of all three search algorithms are within 30% of each other. MD demonstrates a slight advantage not only in overall efficiency but also in the fraction of docked ligands that are within 1 Å RMSD of the crystal structure. The GA comes in second and gives the largest total fraction of docked structures. A single MD energy evaluation is roughly 35% faster than for the MC, due to the smaller nonbonded list cutoff (see Table I). The GA has the slowest energy evaluation (about twice as slow as MD) and provides the smallest number of structures within 1 Å RMSD of the crystal structure (although the most within 3 Å RMSD). Despite performing roughly 40% fewer energy evaluations, the GA gives comparable efficiency to the other search algorithms. Thus, it is probable that, for a grid-based energy function, in which all algorithms would have the same cost per energy evaluation, the GA would outperform other search techniques.

The three algorithms that employ the CHARMM-based energy function are more efficient than AutoDock (using MC simulated annealing). Due to its grid-based energy evaluation, AutoDock performs a single energy evaluation more than ten times faster than the CHARMM based algorithms. However, due to the nature of our designed energy function, despite the lower number of steps, the CHARMM-based docking strategies outperform AutoDock. This observation shows the importance of the energy function in influencing the efficiency of the search strategy.

Table III compares the search algorithms on each complex. The relative performance of the algorithms varies slightly for different complexes. The easiest docking example is benzamidine/trypsin. The most difficult case seems to be sialic acid/hemagglutinin. The poor efficiency for this example is correlated with the least separation between the energy distributions of the docked and misdocked structures (possibly leading to the low fraction of docked structures) and with the largest number of rotatable bonds (giving the

TABLE II.
Comparison of Docking Algorithms.

Algorithm	*SE*	Fraction of docked structures[a]	Fraction of structures with RMSD < 1 Å	ΔE[b]	Time (min)	Number of energy evaluations $\times 10^3$
Small space						
GA	0.41	0.73	0.28	−10.3 (−8.1)	1.8	19
MC	0.31	0.67	0.32	−10.3 (−7.7)	2.1	25
MD	0.29	0.68	0.38	−10.7 (−8.9)	2.3	25
AutoDock	0.16	0.47	0.37	−11.4 (−6.7)	2.8	300
Large space						
GA	0.051	0.58	0.26	−10.6 (−8.2)	11.2	76
MC	0.048	0.49	0.29	−10.6 (−9.1)	10.3	108
MD	0.064	0.49	0.34	−10.5 (−8.2)	7.6	108
AutoDock	0.033	0.36	0.28	−8.7 (−6.3)	11.0	1500

[a] Mean weighted fraction of structures within 3 Å RMSD.
[b] Energy gap between the lowest energy docked structure and the lowest energy misdocked structures; the second number shows the gap between the mean energy of docked structures and the lowest energy misdocked structure.

TABLE III.
Comparison of Algorithms on Different Complexes.

Complex[a]	GA Fraction of docked structures[b]			GA ΔE	GA Time (min)	MC Fraction of docked structures[b]			MC ΔE	MC Time (min)	MD Fraction of docked structures[b]			MD ΔE	MD Time (min)
Small space															
Phosphocholine	0.0	0.9	1.0	−8.9 (−5.7)	1.6	0.0	0.9	1.0	−6.5 (−3.5)	2.0	0.1	0.7	0.8	−7.2 (−4.5)	1.9
Sialic acid	0.0	0.2	0.2	−2.2 (−0.2)	2.9	0.3	0.3	0.3	−2.8 (−0.9)	2.7	0.2	0.3	0.3	−2.5 (−1.6)	3.1
Benzamidine	1.0	1.0	1.0	−26 (−26)	1.4	1.0	1.0	1.0	−26 (−26)	1.9	1.0	1.0	1.0	−26 (−26)	2.2
g3p	0.2	0.8	0.8	−7.5 (−4.1)	1.2	0.2	0.4	0.6	−8.3 (−4.3)	1.8	0.4	0.8	0.9	−8.5 (−6.1)	2.0
Biotin	0.2	0.8	0.8	−4.1 (−3.4)	1.9	0.2	0.3	0.5	−7.3 (−3.0)	2.3	0.3	0.3	0.6	−9.1 (−6.0)	2.5
Large space															
Phosphocholine	0.0	0.8	0.8	−7.5 (−5.5)	6.9	0.1	0.4	0.4	−6.9 (−4.1)	10.6	0.1	0.4	0.6	−7.6 (−4.4)	7.0
Sialic acid	0.1	0.1	0.1	−3.4 (−2.0)	19.5	0.1	0.2	0.2	−3.4 (−3.0)	12.7	0.2	0.3	0.3	−3.5 (−2.5)	10.4
Benzamidine	1.0	1.0	1.0	−26. (−26)	9.4	0.9	0.9	0.9	−26. (−26)	9.6	0.9	0.9	0.9	−26(−26)	7.7
g3p	0.2	0.6	0.6	−7.5 (−3.6)	8.9	0.2	0.6	0.6	−8.5 (−6.3)	8.5	0.3	0.4	0.5	−8.5 (−7.5)	6.8
Biotin	0.1	0.4	0.4	−8.9 (−4.5)	11.5	0.2	0.3	0.3	−8.0 (−3.5)	10.0	0.3	0.3	0.3	−8.0 (−5.4)	7.6

[a] Abbreviations: phosphocholine, phosphocholine / FabMcp603; sialic acid, sialic acid / hemagglutinin; benzamidine, benzamidine / trypsin, g3p, glycerol 3-phosphate; biotin, biotin / streptavidin.
[b] The average fraction of structures with RMSD from crystallographic complex less than 1, 2, and 3 Å, respectively and energies lower then that of the lowest misdocked structure. Other headings coincide with those in Table II.

longest computational times). This example is also the worst case in AutoDock, which finds a very small fraction of docked structures and requires the most computational time. Finally, independent of search algorithm, the efficiency is inversely related to the number of rotatable bonds. The difference in efficiency is not due to the differences in docking time but rather due to the differences in the fraction of the docked structures.

SMALL SEARCH SPACE

A smaller search space requires fewer energy evaluations for exploration, and so we substantially reduced the total number of iterations. This led to shorter computational times and better efficiencies. The total fraction of docked structures is better than for the large search space. The energies of structures with RMSD between 1 Å and 2 Å are, for most complexes, comparable to the energies of structures with RMSD less than 1 Å (especially in the case of phosphocholine). As for the large search space, all algorithms are within 40% of each other in terms of efficiency. In contrast to the large search space, the GA outperforms MC and MD by more than 20%. MD is slightly worse than MC partially due to the longer computational time needed for an energy evaluation (same list cutoff, flexible bond/angle model). As was the case for the large search space, MD gives the best quality of docked structures, whereas the GA gives the largest total fraction of docked structures.

AutoDock, for the small search space is approximately two times less efficient than the other algorithms. The mean time required to perform an energy evaluation is the longest for the GA, due to the necessity of a large nonbonded list. However, the difference is not as striking as for the large search space. Due to the smaller number of energy evaluations, the lowest and mean energies for the smaller search space are higher than for the larger search space. On average, MD provides a slightly lower minimum energy than GA or MC, giving a better overall quality of solutions. As was the case for the large search space, the docking efficiency decreases as the number of rotatable bonds increases. Different algorithms perform better for some complexes. For example, the GA performs better than MC or MD on biotin/streptavidin, while struggling with docking sialic acid to hemagglutinin.

Conclusions

In the accompanying study, we identified an energy function for docking small molecules to their receptors that performed reasonably well.[12]

Using this energy function, we compared three different search algorithms for molecular docking. All algorithms perform reasonably well and are compared in terms of docking efficiency and the energetics of the obtained solutions. We found that our implementation of MD simulated annealing provides the best efficiency in docking structures in the large search space, whereas the GA is the best for the small search space. In addition, regardless of the search space, the MD scheme provides the lowest mean energies of docked structures and structures that are closest to the crystallographic complexes, whereas the GA gives the largest fraction of docked structures. MC simulated annealing performs well for both the large and small search spaces. The efficiency of all our algorithms is more than twice that of AutoDock, which uses a much faster energy evaluation scheme, but falls behind in its ability to find the global minimum. The difference is presumably due to the roughness of the AutoDock energy function and could possibly be improved. The efficiency of our search algorithms increases as the number of degrees of freedom falls, due to the reduced complexity of the search problem.

The computational times for the small search space can be reduced to about 20−30 seconds per ligand on a SGI-R10000 class workstation for rigid ligands (such as benzamidine), while still retaining close to a docking yield of 80%. This indicates that rigid docking can be performed with this model for a small database of more than 8000 compounds in a day on a single workstation. This is still not enough for exhaustive database searches; however, it may be enough to analyze a focused library of lead candidates when an appropriate scoring function is developed for the ranking of these compounds.

Acknowledgments

Fruitful discussions with Professor Andrzej Kolinski and Dr. Leszek Rychlewski are acknowledged. We thank Professor Art Olson and Dr. Garrett Morris for their help with AutoDock.

References

1. G. Jolles and K. R. H. Wooldridge, *Drug Design: Fact or Fantasy?* Academic Press, London, 1984.
2. I. D. Kuntz, *Science*, **257**, 1078 (1992).
3. A. C. Good and J. S. Mason, *Rev. Comput. Chem.*, **7**, 67 (1995).
4. D. A. Gschwend, A. C. Good and I. D. Kuntz, *J. Mol. Recogn.*, **9**, 175 (1996).
5. I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin, *J. Mol. Biol.*, **161**, 269 (1982).
6. D. S. Goodsell and A. J. Olson, *Proteins*, **8**, 195 (1990).
7. A. Caflisch, S. Fischer, and M. Karplus, *J. Comput. Chem.*, **18**, 723 (1997).
8. N. Di Nola, D. Roccatano and H. J. C. Berendsen, *Proteins*, **19**, 174 (1994).
9. D. E. Clark and D. R. Westhead, *J. Comput.-Aided Mol. Design*, **10**, 337 (1996).
10. C. M. Oshiro, I. D. Kuntz, and J. S. Dixon, *J. Comput.-Aided Mol. Design*, **9**, 113 (1995).
11. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comput. Chem.*, **4**, 187 (1983).
12. M. Vieth, J. D. Hirst, A. Kolinski, and C. L. Brooks III, *J. Comput. Chem.* (this issue).
13. J. D. Hirst, B. Dominy, Z. Guo, M. Vieth, and C. L. Brooks III, *Am. Chem. Symp. Series* (in press).
14. D. R. Westhead, D. E. Clark, and C. W. Murray, *J. Comput.-Aided Mol. Design*, **11**, 209 (1997).
15. A. R. Neurath, S. Jiang, K. L. Strick, Y.-Y. Li, and A. K. Debnath, *Nature Med.*, **2**, 230 (1996).
16. P. M. Couthino, M. K. Dowd, and P. J. Reilly, *Proteins*, **28**, 162 (1997).
17. D. E. Koshland Jr., *Proc. Natl. Acad. Sci. USA*, **44**, 98 (1958).
18. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, *J. Mol. Biol.*, **112**, 535 (1977).
19. M. Marquart, J. Walter, J. Deisenhofer, W. Bode, and R. Huber, *Acta Cryst.*, **39**, 480 (1983).
20. E. A. Padlan, G. H. Cohen, and D. R. Davies, *Ann. Immunol.* (*Paris*), **136**, 271 (1985).
21. W. I. Weis, A. T. Bruenger, J. J. Skehel, and D. C. Wiley, *J. Mol. Biol.*, **212**, 737 (1990).
22. M. E. M. Noble, R. K. Wierenga, A. M. Lambeir, F. R. Opperdoes, M. W. H. Thunnissen, K. H. Kalk, H. Groendijk, and W. G. J. Hol, *Proteins*, **10**, 50 (1991).
23. P. C. Weber, D. H. Ohlendorf, J. J. Wendoloski, and F. R. Salemme, *Science*, **243**, 85 (1989).
24. J. M. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
25. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
26. R. S. Judson, E. P. Jaeger, and A. M. Treasurywala, *J. Mol. Struct.*, **308**, 191 (1994).
27. L. D. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Rheinhold, New York, 1991.
28. G. Jones, P. Willet, R. C. Glen, A. R. Leach, and R. Taylor, *J. Mol. Biol.*, **267**, 727 (1997).
29. R. S. Judson, Y. T. Tan, E. Mori, C. Melius, E. P. Jeager, A. M. Treasurywala, and A. Mathiowetz, *J. Comput. Chem.*, **16**, 1405 (1995).